A rewriting system for categorical combinators with multiple arguments

Yokouchi, Hirofumi; Hikita, Teruo

数理解析研究所講究録
第 655 巻 1988 年 186-208

A rewriting system for categorical combinators with multiple arguments

Hirofumi Yokouchi
横内寛文
Tokyo Research Laboratory, IBM Japan
Chiyoda, Tokyo 102, Japan

Teruo Hikita
疋田輝雄
Department of Mathematics, Tokyo Metropolitan University
Setagaya, Tokyo 158, Japan

Version: 26 October 1987

Abstract. Categorical combinators have been derived from the study of categorical semantics of lambda calculus, and it has been found that they may be used in implementation of functional languages. In this paper categorical combinators are extended so that functions with multiple arguments can be directly handled, thus making them more suitable for practical computation. A rewriting system named $CCLM_\beta$ is formulated for these combinators. In this system partial computation is naturally realized. The relationship between this system and lambda calculus is established. As a result of this, the Church-Rosser property of the system is proved.

Key words. Categorical combinator, Church-Rosser property, combinator, functional programming, lambda calculus, partial computation, rewriting system

1. Introduction.

Categorical models of lambda calculus have been extensively studied, e.g. [2], [7], [8], [9], [10], [11], [13]. Curien [4], [5] introduced categorical combinators from such categorical semantics of lambda calculus, and he formulated rewriting systems for them, such as CCL_β and $CCL_{\beta\eta SP}$. Yokouchi [14] independently introduced the CCM calculus, which is equationally equivalent to CCL_β but has slightly different rewriting rules. Incidentally, these systems have a strong resemblance to the functional style language FP of Backus [1]. Categorical combinators have been used in implementation of functional languages [3].

Partial computation, or often called partial evaluation, is a method of computing a function with more than one argument by supplying values to only a specified part of the arguments [6]. It has many applications such as compiler generation.

All the functions associated with the categorical combinators treated so far are presumably one-argument. But this is not directly suited for practical computation where multiple-argument functions are prevailing. Intended for practical application, in this paper a new extended set of categorical combinators is introduced in order to incorporate the notion of functions with multiple arguments. The starting point of the idea is to introduce n-tuples for arbitrary $n \geq 0$, instead of only pairs in the previous categorical combinators. Related to this, we assume that every function in the system has its own fixed *arity*, the number of arguments. With these extensions, the operations of currying and application are naturally extended to "partial currying" and "partial application".

A rewriting system named $CCLM_\beta$ is formulated for these new categorical combinators. One restriction of this system is that n-tuples are not allowed to appear in themselves: they may not appear at "top level", so that tuple-valued functions are not treated. Partial computation is naturally realized in this system. The system is semantically equivalent to type-free λ_β-calculus (without product). Moreover, our results show that these two systems are equivalent even in the sense of reduction. We will establish the natural relationship between the system $CCLM_\beta$ and the lambda calculus by giving translation algorithms between the two. The Church-Rosser property of the system, which is not obtained for the systems CCL_β and $CCL_{\beta\eta SP}$, is proved through this relationship with lambda calculus.

In Section 2 the new categorical combinators are introduced, and the rewriting system $CCLM_\beta$ is formulated for these combinators. A simple example of computation in this system is also given. In Section 3 we briefly state the model theoretic aspect of the system. In Section 4 some derived combinators are introduced which will be useful in practical computation. In Section 5 the translation algorithms are introduced between the $CCLM_\beta$ calculus and lambda calculus, and in Section 6 theorems on the relationship concerning reduction between these two systems are established. Finally, the Church-Rosser property of $CCLM_\beta$ is proved in Section 7, using the results in Section 6.

We assume the reader the basic knowledge of lambda calculus (e.g. [2]). The acquaintance with categorical combinators ([4], [5]) is desirable, but this paper is self-contained and makes no use of previous results about them. We are here mainly interested in the formal aspect of the system, so that application of the categorical combinators with multiple arguments to implementation of functional languages is left to a separate paper.

2. Rewriting system $CCLM_\beta$.

Before presenting the formal system $CCLM_\beta$ of categorical combinators with multiple arguments, we explain the intuitive meaning of each of the new combinators. The combinator $\circ$ means *function composition.* For n functions $f_1, f_2, \ldots, f_n$, $(n \geq 0)$, the angular brackets $\langle f_1, f_2, \ldots, f_n \rangle$ constitute an *n-tuple.* The combinator p_i^n, $1 \leq i \leq n$, is the i-th *projection* of an n-tuple.

We extend the usual *currying* operation to functions with n arguments. For $n \geq 1$, the combinator Λ_n applies to a function with n arguments and means currying. More precisely, for an n-argument function f, $\Lambda_n(f)$ is an $(n-1)$-argument function whose arguments correspond to the first $n-1$ arguments of f. The resulting value of the function $\Lambda_n(f)$ is a one-argument function, whose argument corresponds to the last argument of f.

We also extend the usual combinator of function application *App* to *partial application.* In our definition *App* receives two arguments, an n-argument function and a value; it implies applying the value as the first argument among n arguments of the function, and returns a function with $n-1$ arguments.

Now we formally give the definition of terms of $CCLM_\beta$.

Definition. We define *terms* with nonnegative integers called *arity.* In the following, terms are denoted by capital letters F, G, F_1, etc. For every constant its arity is uniquely specified. We assume that there are special constants: p_i^n of arity n where $n \geq 1$ and $1 \leq i \leq n$, and *App* of arity 2. Then the terms are defined inductively as follows.
(1) every constant is a term.
(2) if F is a term of arity m and $G_1, \ldots, G_m$ are terms of arity n, $m \geq 0$, $n \geq 0$, then $F \circ \langle G_1, \ldots, G_m \rangle^n$ is a term of arity n.
(3) if F is a term of arity n, $n \geq 1$, then $\Lambda_n(F)$ is a term of arity $n-1$.

We omit the superscript n of $F \circ \langle G_1, \ldots, G_m \rangle^n$ whenever no confusion occurs. Also, we often omit the subscript of $\Lambda_n(F)$ like $\Lambda(F)$.

Note that when F is a term of arity 0, $F \circ \langle\ \rangle^n$ is a term of arity n. Note also that in this system n-tuples in themselves are *not* terms; they always appear as part of composed terms.

Now we present the rewriting rules of the formal system $CCLM_\beta$.

Definition. We define the binary relation $\to$ among the terms of $CCLM_\beta$ by the following rules:

1. $(F\circ\langle G_1,\dots,G_m\rangle^n)\circ\langle H_1,\dots,H_n\rangle^k \to F\circ\langle G_1\circ\langle H_1,\dots,H_n\rangle^k,\dots,G_m\circ\langle H_1,\dots,H_n\rangle^k\rangle^k$.
2. $p_i^n\circ\langle F_1,\dots,F_n\rangle \to F_i$.
3. $F\circ\langle p_1^n,\dots,p_n^n\rangle \to F$,
   where F is of arity n.
4. $\Lambda_{n+1}(F)\circ\langle G_1,\dots,G_n\rangle^k$
   $$\to \Lambda_{k+1}(F\circ\langle G_1\circ\langle p_1^{k+1},\dots,p_k^{k+1}\rangle,\dots,G_n\circ\langle p_1^{k+1},\dots,p_k^{k+1}\rangle,p_{k+1}^{k+1}\rangle^{k+1}),$$
   where F is of arity $(n+1)$, and $G_1,\dots,G_n$ are of arity k.
5. $App\circ\langle\Lambda_{n+1}(F),G\rangle^n \to F\circ\langle p_1^n,\dots,p_n^n,G\rangle^n$,
   where F is of arity $n+1$, and G is of arity n.
6. If $F\to F'$, then $F\circ\langle G_1,\dots,G_m\rangle \to F'\circ\langle G_1,\dots,G_m\rangle$.
7. If $G_i\to G_i'$ for some $1\le i\le m$, then $F\circ\langle G_1,\dots,G_i,\dots,G_m\rangle$
   $\to F\circ\langle G_1,\dots,G_i',\dots,G_m\rangle$.
8. If $F\to F'$, then $\Lambda(F)\to\Lambda(F')$.

We sometimes denote this relation $\to$ by $\to_c$, especially when it is necessary to distinguish it from that of lambda calculus ($\to_\lambda$). We denote by $\xrightarrow{*}$ the reflexive and transitive closure of $\to$. Note that arity is invariant under the relation $\to$ (and $\xrightarrow{*}$).

Example. Computation in $CCLM_\beta$.
Let $plus(x,y,z)=x+y+z$ be a function with 3 arguments giving their sum. In $CCLM_\beta$ this is translated to the following (the translation algorithm will be given in Section 5):
$$\Lambda_1(\Lambda_2(\Lambda_3(plus\circ\langle p_1^3,p_2^3,p_3^3\rangle))).$$

Now, we give only one value 2 to its first argument, and partially compute it using *App*. In the below, 2^n means the constant-valued function with n arguments giving 2 as its result.

$$\begin{array}{ll} App\circ\langle\Lambda_1(\Lambda_2(\Lambda_3(plus\circ\langle p_1^3,p_2^3,p_3^3\rangle))),2^0\rangle & \\ \to \Lambda_2(\Lambda_3(plus\circ\langle p_1^3,p_2^3,p_3^3\rangle))\circ\langle 2^0\rangle & \text{(by rule 5)} \\ \to \Lambda_1(\Lambda_3(plus\circ\langle p_1^3,p_2^3,p_3^3\rangle)\circ\langle 2^0\circ\langle\,\rangle^1,p_1^1\rangle) & \text{(by rule 4)} \\ \to \Lambda_1(\Lambda_3(plus\circ\langle p_1^3,p_2^3,p_3^3\rangle)\circ\langle 2^1,p_1^1\rangle) & \end{array}$$

$$\rightarrow \Lambda_1(\Lambda_2((plus \circ \langle p_1^3, p_2^3, p_3^3\rangle) \circ \langle 2^1 \circ \langle p_1^2\rangle, p_1^1 \circ \langle p_1^2\rangle, p_2^2\rangle)) \qquad \text{(by rules 4, 8)}$$
$$\xrightarrow{*} \Lambda_1(\Lambda_2((plus \circ \langle p_1^3, p_2^3, p_3^3\rangle) \circ \langle 2^2, p_1^2, p_2^2\rangle))$$
$$\xrightarrow{*} \Lambda_1(\Lambda_2(plus \circ \langle 2^2, p_1^2, p_2^2\rangle)).$$

3. On models of $CCLM_\beta$.

Before we examine the properties of $CCLM_\beta$ as a rewriting system, we digress and make a brief discussion about models of $CCLM_\beta$ as an equational system. Those who are interested only in the operational aspect of the system may skip this section.

Let **C** be a Cartesian closed category (ccc). We say that an object u of **C** is *reflexive*, when there exists a pair of arrows $\phi\colon u \to u^u$ and $\psi\colon u^u \to u$ such that $\phi \circ \psi = id_{u^u}$. It is known that ccc's with reflexive object are essentially the same as models of lambda calculus. See [2], [5], [8], [10], [11]. Similarly, ccc's with reflexive object characterize models of $CCLM_\beta$. We can naturally interpret terms of $CCLM_\beta$ in a ccc **C** with reflexive object u. Terms of arity n in $CCLM_\beta$ are interpreted in the set $\mathbf{C}(u^n, u)$ of arrows from u^n to u. Here u^n denotes the product

$$t \times u \times \cdots \times u \qquad (n \text{ times of } u),$$

where t is the terminal object of **C**.

More formally, the interpretation of terms in **C** is the following. For each term F of arity n, we define the arrow $[\![F]\!]$ from u^n to u in **C** as follows. Here we assume that for every constant f other than p_i^n and App, $[\![f]\!]$ is already specified.

1. $[\![p_i^n]\!] = \pi_{i+1}^{t,u,\ldots,u}$ (the $(i+1)$-th projection from u^n to u).
2. $[\![App]\!] = ev^{u,u} \circ (\phi \times id_u)$,
   where $ev^{u,u}$ is the evaluation map from u^u to u.
3. $[\![F \circ \langle G_1, \ldots, G_n\rangle]\!] = [\![F]\!] \circ \langle [\![G_1]\!], \ldots, [\![G_n]\!]\rangle$.
4. $[\![\Lambda(F)]\!] = \psi \circ \Lambda_{u^n,u}([\![F]\!] \circ \pi_1^{u^n,u})$,

where $\Lambda_{u^n,u}(h): u^n \to u^u$ is the transpose map of $h: u^n \times u \to u$.

Based on this interpretation, we can prove that, if $F = G$ in $CCLM_\beta$ as an equational system, then $[\![F]\!] = [\![G]\!]$ in $\mathbf{C}$. That is, $\mathbf{C}$ is a model of $CCLM_\beta$.

4. Auxiliary combinators.

We further introduce the derived combinators of currying and application in a more general form. First the intuitive meanings. For $m \geq 1$, the operator $\Lambda^m(-)$ means currying m times. Thus, for an n-argument function f, $\Lambda_n^m(f)$ is an $(n-m)$-argument function which gives an m-argument function as a result. More precisely,

$$\Lambda_n^m(f) \equiv \Lambda_{n-m+1}(\Lambda_{n-m+2}(\cdots(\Lambda_n(f))\cdots)).$$

Informally, in a lambda-calculus like notation, $\Lambda_n^m(f)$ means

$$\lambda\langle x_1, \ldots, x_{n-m}\rangle.(\lambda x_{n-m+1} \ldots \lambda x_n.f\langle x_1, \ldots, x_n\rangle).$$

(We do not yet, and will not in this paper, formally define the angular brackets $\langle\ \rangle$ in lambda calculus. The above expression is only for the reader's intuitive understanding.)

Likewise, for $m \geq 1$, App^m receives $m+1$ arguments, an n-argument function, $(n \geq m)$, and m values; it implies applying the values as the first m arguments of the function, and returns a function of $n-m$ arguments. Thus, in particular, for an n-argument function, App^n is the usual (full) application. App^m is informally represented by

$$\lambda\langle z, x_1, \ldots, x_m\rangle.zx_1 \ldots x_m.$$

Now we formally introduce auxiliary combinators.

(1) For $n \geq 0$, define $id^n \equiv \langle p_1^n, \ldots, p_n^n\rangle^n$.
(2) For $m \geq 0$ and $n \geq 0$, define $\pi_1^{m,n} \equiv \langle p_1^{m+n}, \ldots, p_m^{m+n}\rangle^{m+n}$, and $\pi_2^{m,n} \equiv \langle p_{m+1}^{m+n}, \ldots, p_{m+n}^{m+n}\rangle^{m+n}$.
(3) For $n \geq 1$, $1 \leq m \leq n$, and F a term of arity n, define the $(n-m)$-ary term $\Lambda^m(F)$ inductively by $\Lambda^1(F) \equiv \Lambda(F)$, and $\Lambda^{m+1}(F) \equiv \Lambda(\Lambda^m(F))$.
(4) For $m \geq 1$ define

$$App^m \equiv App \circ \langle App \circ \langle \cdots \langle App \circ \langle App \circ \langle p_1^{m+1}, p_2^{m+1}\rangle, p_3^{m+1}\rangle, p_4^{m+1}\rangle, \cdots\rangle, p_{m+1}^{m+1}\rangle,$$

(m times of App).

Remark 1. For $m \geq 1$ and $n \geq 1$,

$$\Lambda^m(\Lambda^n(F)) \equiv \Lambda^{m+n}(F).$$

Remark 2.
For $m \geq 1$ and n-ary terms $F, G_1, \ldots, G_m$, define the n-ary term $APP^m\{F, G_1, \ldots, G_m\}$ inductively by

$$APP^1\{F, G_1\} \equiv App \circ \langle F, G_1\rangle,$$

and

$$APP^{m+1}\{F, G_1, \ldots, G_{m+1}\} \equiv App \circ \langle APP^m\{F, G_1, \ldots, G_m\}, G_{m+1}\rangle.$$

Then we easily have:

i) $\quad App^m \equiv APP^m\{p_1^{m+1}, p_2^{m+1}, \ldots, p_{m+1}^{m+1}\}$,

ii) $\quad App^m \circ \langle F, G_1, \ldots, G_m\rangle \xrightarrow{*} APP^m\{F, G_1, \ldots, G_m\}$,

iii) $\quad APP^m\{APP^l\{F, G_1, \ldots, G_l\}, H_1, \ldots, H_m\} \equiv APP^{l+m}\{F, G_1, \ldots, G_l, H_1, \ldots, H_m\}$.

Rules 4 and 5 of $CCLM_\beta$ have natural extensions for Λ^m and App^m, which are the following propositions.

Proposition 4.1. *Let F be $(m+n)$-ary and $G_1, \ldots, G_n$ be k-ary, where $m \geq 1$. Then*

$$\Lambda^m_{m+n}(F) \circ \langle G_1, \ldots, G_n\rangle^k$$
$$\xrightarrow{*} \Lambda^m_{k+m}(F \circ \langle G_1 \circ \pi_1^{k,m}, \ldots, G_n \circ \pi_1^{k,m}, p_{k+1}^{k+m}, \ldots, p_{k+m}^{k+m}\rangle^{k+m}).$$

Proof. Induction on m. When $m = 1$ this is identical to rule 4 of $CCLM_\beta$.

$$\Lambda^{m+1}(F) \circ \langle G_1, \ldots, G_n\rangle$$
$$\to \Lambda(\Lambda^m(F) \circ \langle G_1 \circ \pi_1^{k,1}, \ldots, G_n \circ \pi_1^{k,1}, p_{k+1}^{k+1}\rangle) \qquad \text{(by rule 4)}$$
$$\xrightarrow{*} \Lambda(\Lambda^m(F \circ \langle (G_1 \circ \pi_1^{k,1}) \circ \pi_1^{k+1,m}, \ldots, (G_n \circ \pi_1^{k,1}) \circ \pi_1^{k+1,m}, p_{k+1}^{k+1} \circ \pi_1^{k+1,m},$$
$$p_{k+2}^{k+1+m}, \ldots, p_{k+1+m}^{k+1+m}\rangle)) \qquad \text{(by induction hypothesis)}$$
$$\xrightarrow{*} \Lambda^{m+1}(F \circ \langle G_1 \circ \pi_1^{k,m+1}, \ldots, G_n \circ \pi_1^{k,m+1}, p_{k+1}^{k+m+1}, \ldots, p_{k+m+1}^{k+m+1}\rangle).$$

Proposition 4.2. *Let F be $(m+n)$-ary, and $G_1, \ldots, G_m$ be n-ary, where $m \geq 1$. Then,*

$$APP^m\{\Lambda^m_{m+n}(F), G_1, \ldots, G_m\} \xrightarrow{*} F \circ \langle p_1^n, \ldots, p_n^n, G_1, \ldots, G_m\rangle^n.$$

Proof. By induction on m. When $m = 1$ this is rule 5.

$$APP^{m+1}\{\Lambda^{m+1}(F), G_1, \ldots, G_m, G_{m+1}\}$$
$$\equiv App \circ \langle APP^m\{\Lambda^{m+1}(F), G_1, \ldots, G_m\}, G_{m+1}\rangle$$
$$\xrightarrow{*} App \circ \langle \Lambda(F) \circ \langle p_1^n, \ldots, p_n^n, G_1, \ldots, G_m\rangle, G_{m+1}\rangle \qquad \text{(by induction hypothesis)}$$

$$\begin{aligned}
&\to App \circ \langle \Lambda(F \circ \langle p_1^n \circ \pi_1^{n,1}, \ldots, p_n^n \circ \pi_1^{n,1}, G_1 \circ \pi_1^{n,1}, \ldots, G_m \circ \pi_1^{n,1}, p_{n+1}^{n+1}\rangle), G_{m+1}\rangle \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(by rules 4, 7)} \\
&\xrightarrow{*} App \circ \langle \Lambda(F \circ \langle p_1^{n+1}, \ldots, p_n^{n+1}, G_1 \circ \pi_1^{n,1}, \ldots, G_m \circ \pi_1^{n,1}, p_{n+1}^{n+1}\rangle), G_{m+1}\rangle \\
&\to (F \circ \langle p_1^{n+1}, \ldots, p_n^{n+1}, G_1 \circ \pi_1^{n,1}, \ldots, G_m \circ \pi_1^{n,1}, p_{n+1}^{n+1}\rangle) \circ \langle p_1^n, \ldots, p_n^n, G_{m+1}\rangle \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(by rule 5)} \\
&\xrightarrow{*} F \circ \langle p_1^n, \ldots, p_n^n, G_1, \ldots, G_m, G_{m+1}\rangle.
\end{aligned}$$

Proposition 4.3. *Let F be $(m+n)$-ary, and $G_1, \ldots, G_m$ be n-ary, where $m \geq 1$. Then,*

$$App^m \circ \langle \Lambda^m_{m+n}(F), G_1, \ldots, G_m\rangle^n \xrightarrow{*} F \circ \langle p_1^n, \ldots, p_n^n, G_1, \ldots, G_m\rangle^n.$$

Proof. Immediate by combining ii) of remark 2 above and Proposition 4.2.

The auxiliary combinators will be useful in actual computations in $CCLM_\beta$, since, as the example below indicates, they can be used to shorten the length of computation.

Example. Computation in $CCLM_\beta$ with the auxiliary combinators.
Let us use the same function $plus(x, y, z) = x + y + z$, and give two values 2 and 3 to it:

$$\begin{aligned}
&App^2 \circ \langle \Lambda^3(plus \circ \langle p_1^3, p_2^3, p_3^3\rangle), 2^0, 3^0\rangle \\
&\xrightarrow{*} \Lambda(plus \circ \langle p_1^3, p_2^3, p_3^3\rangle) \circ \langle 2^0, 3^0\rangle && \text{(by Proposition 4.3)} \\
&\to \Lambda((plus \circ \langle p_1^3, p_2^3, p_3^3\rangle) \circ \langle 2^0 \circ \langle\,\rangle^1, 3^0 \circ \langle\,\rangle^1, p_1^1\rangle) && \text{(by rule 4)} \\
&\xrightarrow{*} \Lambda(plus \circ \langle 2^1, 3^1, p_1^1\rangle).
\end{aligned}$$

5. Translations between $CCLM_\beta$ and lambda calculus.

In this section we define translation algorithms for both directions between $CCLM_\beta$ and lambda calculus, and we establish the natural relationship between the terms in these two systems. The lambda calculus we are concerned is, more specifically, the type-free λ_β-calculus (without product), which we will denote by λ. We assume that λ contains constants. Moreover, we assume that there is given a one-to-one correspondence between the constants in $CCLM_\beta$ other than p_i^n and App, and the constants in λ.

Firstly, the translation algorithm from λ to $CCLM_\beta$ is described. For terms M and N in λ, we denote by $M[x := N]$ the λ-term obtained by substituting N for each occurrence of a free variable x in M.

Convention. Let $\gamma \equiv \langle z_1, \ldots, z_n \rangle$ be a sequence of distinct variables $z_1, \ldots, z_n$, $n \geq 0$. For such a sequence γ and a variable x, we denote by γx the sequence of the elements of γ followed by x, that is, $\gamma x \equiv \langle z_1, \ldots, z_n, x \rangle$. Similarly, for two sequences $\alpha \equiv \langle x_1, \ldots, x_l \rangle$ and $\beta \equiv \langle y_1, \ldots, y_m \rangle$, we denote $\alpha\beta \equiv \langle x_1, \ldots, x_l, y_1, \ldots, y_m \rangle$.

Definition. For each term M in λ whose free variables are contained in $\gamma \equiv \langle z_1, \ldots, z_n \rangle$, we define inductively a term of arity n in $CCLM_\beta$, denoted by $[\lambda\gamma.M]$, as follows:

1. $[\lambda\gamma.z_i] \equiv p_i^n$, $1 \leq i \leq n$.
2. $[\lambda\gamma.c] \equiv \Lambda^s(c \circ \langle p_{n+1}^{n+s}, \ldots, p_{n+s}^{n+s} \rangle)$,
   where c is a constant of arity s, $s \geq 1$. When $s = 0$, we define $[\lambda\gamma.c] \equiv c \circ \langle\ \rangle^n$.
3. $[\lambda\gamma.(\lambda x.M)] \equiv \Lambda([\lambda\gamma x'.M[x := x']])$,
   where $x' \equiv x$ if x is not in γ, otherwise x' is a new variable.
4. $[\lambda\gamma.cM_1 \ldots M_s] \equiv c \circ \langle [\lambda\gamma.M_1], \ldots, [\lambda\gamma.M_s] \rangle^n$,
   where c is a constant of arity s, $s \geq 0$.
5. $[\lambda\gamma.M_1M_2] \equiv App \circ \langle [\lambda\gamma.M_1], [\lambda\gamma.M_2] \rangle$,
   where M_1M_2 is not of the form in 4.

In the following discussions, whenever we mention $[\lambda\langle z_1, \ldots, z_n\rangle.M]$, we assume that the variables $z_1, \ldots, z_n$ are distinct and that all the free variables in M are contained in the set $\{z_1, \ldots, z_n\}$.

Remark. The above definition of translation for constants of case 4 may seem somewhat artificial. Indeed, a different definition of translation without case 4 (that is, case 2 only) would be simpler, and actually almost sufficient. Under this simpler definition, the righthand-side of the translation of case 4 is obtained as a result of reductions for the translated term of the lefthand-side (inspect the proof of Proposition 5.1 below). However, in Theorem 6.3 of the next section (precisely, in case 3 of its proof), we need our present translation for constants.

We give the general case of the translation of constants from λ to $CCLM_\beta$.

Proposition 5.1. *Let c be an s-ary constant in λ, and $0 < m < s$. Let $\gamma \equiv \langle z_1, \ldots, z_n \rangle$. Then we have:*

$$
\begin{aligned}
&[\lambda\gamma.cM_1 \ldots M_m] \\
&\xrightarrow{*}_c \Lambda^{s-m}(c \circ \langle [\lambda\gamma.M_1] \circ \pi_1^{n,s-m}, \ldots, [\lambda\gamma.M_m] \circ \pi_1^{n,s-m}, p_{n+1}^{n+s-m}, \ldots, p_{n+s-m}^{n+s-m} \rangle).
\end{aligned}
$$

Proof.

$$
\begin{aligned}
&[\lambda\gamma.cM_1 \ldots M_m] \\
&\equiv App \circ \langle App \circ \langle \cdots \langle App \circ \langle [\lambda\gamma.c], [\lambda\gamma.M_1] \rangle, [\lambda\gamma.M_2] \rangle \cdots \rangle, [\lambda\gamma.M_m] \rangle \\
&\equiv APP^m\{[\lambda\gamma.c], [\lambda\gamma.M_1], \ldots, [\lambda\gamma.M_m]\} \\
&\equiv APP^m\{\Lambda^s(c \circ \pi_2^{n,s}), [\lambda\gamma.M_1], \ldots, [\lambda\gamma.M_m]\} \\
&\xrightarrow{*} \Lambda^{s-m}(c \circ \pi_2^{n,s}) \circ \langle p_1^n, \ldots, p_n^n, [\lambda\gamma.M_1], \ldots, [\lambda\gamma.M_m] \rangle \qquad \text{(by Proposition 4.2)} \\
&\xrightarrow{*} \Lambda^{s-m}((c \circ \pi_2^{n,s}) \circ \langle p_1^n \circ \pi_1^{n,s-m}, \ldots, p_n^n \circ \pi_1^{n,s-m}, \\
&\qquad [\lambda\gamma.M_1] \circ \pi_1^{n,s-m}, \ldots, [\lambda\gamma.M_m] \circ \pi_1^{n,s-m}, p_{n+1}^{n+s-m}, \ldots, p_{n+s-m}^{n+s-m} \rangle) \\
&\qquad \text{(by Proposition 4.1)} \\
&\xrightarrow{*} \Lambda^{s-m}(c \circ \langle [\lambda\gamma.M_1] \circ \pi_1^{n,s-m}, \ldots, [\lambda\gamma.M_m] \circ \pi_1^{n,s-m}, p_{n+1}^{n+s-m}, \ldots, p_{n+s-m}^{n+s-m} \rangle).
\end{aligned}
$$

Remark. If we use Lemma 5.3 which will be established soon, the righthand-side of the rewriting of the above proposition can be transformed further:

$$
\begin{aligned}
&\text{(the last term of the proof)} \\
&\xrightarrow{*} \Lambda^{s-m}(c \circ \langle [\lambda\gamma x_1 \ldots x_{s-m}.M_1], \ldots, [\lambda\gamma x_1 \ldots x_{s-m}.M_m], \\
&\qquad [\lambda\gamma x_1 \ldots x_{s-m}.x_1], \ldots, [\lambda\gamma x_1 \ldots x_{s-m}.x_{s-m}] \rangle) \qquad \text{(by Lemma 5.3)} \\
&\equiv \Lambda^{s-m}([\lambda\gamma x_1 \ldots x_{s-m}.cM_1 \ldots M_m x_1 \ldots x_{s-m}]).
\end{aligned}
$$

Next, we give the translation algorithm from $CCLM_\beta$ to λ.

Definition. For each term F of arity n in $CCLM_\beta$ and terms $N_1, \ldots, N_n$ of λ we define the term $F^*[N_1, \ldots, N_n]$ in λ, inductively by the structure of F, as follows.

1. $(p_i^n)^*[N_1, \ldots, N_n] \equiv N_i$.
2. $App^*[N_1, N_2] \equiv N_1 N_2$.
3. $f^*[N_1, \ldots, N_n] \equiv f N_1 \ldots N_n$,
   for each n-ary constant f other than p_i^n and App.

4. $(F \circ \langle G_1, \ldots, G_m \rangle^n)^*[N_1, \ldots, N_n] \equiv F^*[G_1^*[N_1, \ldots, N_n], \ldots, G_m^*[N_1, \ldots, N_n]]$.
5. $(\Lambda(F))^*[N_1, \ldots, N_n] \equiv \lambda x.(F^*[N_1, \ldots, N_n, x])$,
   where x is a variable not free in $N_1, \ldots, N_n$.

A term F of arity n in $CCLM_\beta$ means an n-ary function. Thus F is intuitively represented by a λ-term M with free variables $x_1, \ldots, x_n$. In the above definition $F^*[N_1, \ldots, N_n]$ means $M[x_1 := N_1, \ldots, x_n := N_n]$.

Now we return to the former translation algorithm and give three basic lemmas concerning it.

Lemma 5.2. $[\lambda\langle z_1, \ldots, z_n\rangle.M] \equiv [\lambda\langle z'_1, \ldots, z'_n\rangle.M[z_1 := z'_1, \ldots, z_n := z'_n]]$.

Proof. Easy and omitted.

Lemma 5.3. *Let* $\alpha = \langle x_1, \ldots, x_l\rangle$, $\beta = \langle y_1, \ldots, y_m\rangle$, *and* $\gamma = \langle z_1, \ldots, z_n\rangle$. *Then,*

$$[\lambda\alpha\gamma.M] \circ \langle p_1^{l+m+n}, \ldots, p_l^{l+m+n}, p_{l+m+1}^{l+m+n}, \ldots, p_{l+m+n}^{l+m+n}\rangle \xrightarrow{*}_c [\lambda\alpha\beta\gamma.M].$$

As a special case, we have

$$[\lambda\alpha.M] \circ \langle p_1^{l+m}, \ldots, p_l^{l+m}\rangle \xrightarrow{*}_c [\lambda\alpha\beta.M].$$

Proof. The proof is by induction on the structure of M.

Case 1. $M \equiv x_i$, $1 \leq i \leq l$.

$$\begin{aligned} &[\lambda\alpha\gamma.x_i] \circ \langle p_1^{l+m+n}, \ldots, p_l^{l+m+n}, p_{l+m+1}^{l+m+n}, \ldots, p_{l+m+n}^{l+m+n}\rangle \\ &\equiv p_i^{l+n} \circ \langle p_1^{l+m+n}, \ldots, p_l^{l+m+n}, p_{l+m+1}^{l+m+n}, \ldots, p_{l+m+n}^{l+m+n}\rangle \\ &\to p_i^{l+m+n} \\ &\equiv [\lambda\alpha\beta\gamma.x_i]. \end{aligned}$$

Case 2. $M \equiv z_k$, $1 \leq k \leq n$.
Similar to case 1.
Case 3. $M \equiv c$ (s-ary constant).
When $s = 0$ it is clear. Suppose $s \geq 1$.

$$\begin{aligned} &[\lambda\alpha\gamma.c] \circ \langle p_1^{l+m+n}, \ldots, p_l^{l+m+n}, p_{l+m+1}^{l+m+n}, \ldots, p_{l+m+n}^{l+m+n}\rangle \\ &\equiv \Lambda^s(c \circ \pi_2^{l+n,s}) \circ \langle p_1^{l+m+n}, \ldots, p_l^{l+m+n}, p_{l+m+1}^{l+m+n}, \ldots, p_{l+m+n}^{l+m+n}\rangle \\ &\xrightarrow{*} \Lambda^s((c \circ \pi_2^{l+n,s}) \circ \langle p_1^{l+m+n} \circ \pi_1^{l+m+n,s}, \ldots, p_l^{l+m+n} \circ \pi_1^{l+m+n,s}, \\ &\qquad p_{l+m+1}^{l+m+n} \circ \pi_1^{l+m+n,s}, \ldots, p_{l+m+n}^{l+m+n} \circ \pi_1^{l+m+n,s}, p_{l+m+n+1}^{l+m+n+s}, \ldots, p_{l+m+n+s}^{l+m+n+s}\rangle) \\ &\qquad\qquad \text{(by Proposition 4.1)} \\ &\xrightarrow{*} \Lambda^s(c \circ \pi_2^{l+m+n,s}) \\ &\equiv [\lambda\alpha\beta\gamma.c]. \end{aligned}$$

Case 4. $M \equiv \lambda w.M_1$.

$$
\begin{aligned}
&[\lambda\alpha\gamma.(\lambda w.M_1)] \circ \langle p_1^{l+m+n}, \ldots, p_l^{l+m+n}, p_{l+m+1}^{l+m+n}, \ldots, p_{l+m+n}^{l+m+n}\rangle \\
&\equiv \Lambda([\lambda\alpha\gamma w'.M_1[w := w']]) \circ \langle p_1^{l+m+n}, \ldots, p_l^{l+m+n}, p_{l+m+1}^{l+m+n}, \ldots, p_{l+m+n}^{l+m+n}\rangle \\
&\rightarrow \Lambda([\lambda\alpha\gamma w'.M_1[w := w']] \circ \langle p_1^{l+m+n} \circ \pi_1^{l+m+n,1}, \ldots, p_l^{l+m+n} \circ \pi_1^{l+m+n,1}, \\
&\qquad p_{l+m+1}^{l+m+n} \circ \pi_1^{l+m+n,1}, \ldots, p_{l+m+n}^{l+m+n} \circ \pi_1^{l+m+n,1}, p_{l+m+n+1}^{l+m+n+1}\rangle) \\
&\qquad\qquad \text{(by rule 4)} \\
&\xrightarrow{*} \Lambda([\lambda\alpha\gamma w'.M_1[w := w']] \circ \langle p_1^{l+m+n+1}, \ldots, p_l^{l+m+n+1}, \\
&\qquad p_{l+m+1}^{l+m+n+1}, \ldots, p_{l+m+n}^{l+m+n+1}, p_{l+m+n+1}^{l+m+n+1}\rangle) \\
&\xrightarrow{*} \Lambda([\lambda\alpha\beta\gamma w'.M_1[w := w']]) \qquad \text{(by induction hypothesis)} \\
&\equiv [\lambda\alpha\beta\gamma.(\lambda w.M_1)].
\end{aligned}
$$

Case 5. $M \equiv cM_1 \ldots M_s$ (c is an s-ary constant).

$$
\begin{aligned}
&[\lambda\alpha\gamma.cM_1 \ldots M_s] \circ \langle p_1^{l+m+n}, \ldots, p_l^{l+m+n}, p_{l+m+1}^{l+m+n}, \ldots, p_{l+m+n}^{l+m+n}\rangle \\
&\equiv (c \circ \langle[\lambda\alpha\gamma.M_1], \ldots, [\lambda\alpha\gamma.M_s]\rangle) \circ \langle p_1^{l+m+n}, \ldots, p_l^{l+m+n}, p_{l+m+1}^{l+m+n}, \ldots, p_{l+m+n}^{l+m+n}\rangle \\
&\rightarrow c \circ \langle[\lambda\alpha\gamma.M_1] \circ \langle p_1^{l+m+n}, \ldots, p_l^{l+m+n}, p_{l+m+1}^{l+m+n}, \ldots, p_{l+m+n}^{l+m+n}\rangle, \ldots, \\
&\qquad [\lambda\alpha\gamma.M_s] \circ \langle p_1^{l+m+n}, \ldots, p_l^{l+m+n}, p_{l+m+1}^{l+m+n}, \ldots, p_{l+m+n}^{l+m+n}\rangle\rangle \\
&\qquad\qquad \text{(by rule 1)} \\
&\xrightarrow{*} c \circ \langle[\lambda\alpha\beta\gamma.M_1], \ldots, [\lambda\alpha\beta\gamma.M_s]\rangle \qquad \text{(by induction hypothesis)} \\
&\equiv [\lambda\alpha\beta\gamma.cM_1 \ldots M_s].
\end{aligned}
$$

Case 6. $M \equiv M_1M_2$, and M is not of the form of case 5.

$$
\begin{aligned}
&[\lambda\alpha\gamma.M_1M_2] \circ \langle p_1^{l+m+n}, \ldots, p_l^{l+m+n}, p_{l+m+1}^{l+m+n}, \ldots, p_{l+m+n}^{l+m+n}\rangle \\
&\equiv (App \circ \langle[\lambda\alpha\gamma.M_1], [\lambda\alpha\gamma.M_2]\rangle) \circ \langle p_1^{l+m+n}, \ldots, p_l^{l+m+n}, p_{l+m+1}^{l+m+n}, \ldots, p_{l+m+n}^{l+m+n}\rangle \\
&\xrightarrow{*} App \circ \langle[\lambda\alpha\beta\gamma.M_1], [\lambda\alpha\beta\gamma.M_2]\rangle \qquad \text{(by induction hypothesis)} \\
&\equiv [\lambda\alpha\beta\gamma.M_1M_2].
\end{aligned}
$$

Lemma 5.4. *Let* $\alpha = \langle x_1, \ldots, x_m\rangle$, and $\gamma = \langle z_1, \ldots, z_n\rangle$. *Then*

$$[\lambda\alpha.M] \circ \langle[\lambda\gamma.N_1], \ldots, [\lambda\gamma.N_m]\rangle \xrightarrow{*}_c [\lambda\gamma.M[x_1 := N_1, \ldots, x_m := N_m]].$$

Proof. By induction on the structure of M.

Case 1. $M \equiv x_i$, $1 \le i \le m$.

$$
\begin{aligned}
&[\lambda\alpha.x_i] \circ \langle[\lambda\gamma.N_1], \ldots, [\lambda\gamma.N_m]\rangle \\
&\equiv p_i^m \circ \langle[\lambda\gamma.N_1], \ldots, [\lambda\gamma.N_m]\rangle \\
&\rightarrow [\lambda\gamma.N_i].
\end{aligned}
$$

Case 2. $M \equiv c$ (c is an s-ary constant).

When $s = 0$ it is clear. Suppose $s \ge 1$.

$$
\begin{aligned}
&[\lambda\alpha.c] \circ \langle[\lambda\gamma.N_1], \ldots, [\lambda\gamma.N_m]\rangle \\
&\equiv \Lambda^s(c \circ \pi_2^{m,s}) \circ \langle[\lambda\gamma.N_1], \ldots, [\lambda\gamma.N_m]\rangle \\
&\xrightarrow{*} \Lambda^s((c \circ \pi_2^{m,s}) \circ \langle[\lambda\gamma.N_1] \circ \pi_1^{n,s}, \ldots, [\lambda\gamma.N_m] \circ \pi_1^{n,s}, p_{n+1}^{n+s}, \ldots, p_{n+s}^{n+s}\rangle)
\end{aligned}
$$

(by Proposition 4.1)

$\xrightarrow{*} \Lambda^s(c \circ \pi_2^{n,s})$

$\equiv [\lambda\gamma.c]$.

Case 3. $M \equiv \lambda x.M_1$.

$[\lambda\alpha.(\lambda x.M_1)] \circ \langle[\lambda\gamma.N_1], \ldots, [\lambda\gamma.N_m]\rangle$

$\equiv \Lambda([\lambda\alpha x'.M_1[x := x']]) \circ \langle[\lambda\gamma.N_1], \ldots, [\lambda\gamma.N_m]\rangle$

$\rightarrow \Lambda([\lambda\alpha x'.M_1[x := x']] \circ \langle[\lambda\gamma.N_1] \circ \pi_1^{n,1}, \ldots, [\lambda\gamma.N_m] \circ \pi_1^{n,1}, p_{n+1}^{n+1}\rangle)$

$\xrightarrow{*} \Lambda([\lambda\alpha x'.M_1[x := x']] \circ \langle[\lambda\gamma x'.N_1], \ldots, [\lambda\gamma x'.N_m], [\lambda\gamma x'.x']\rangle)$

(by Lemma 5.3)

$\xrightarrow{*} \Lambda([\lambda\gamma x'.M_1[x_1 := N_1, \ldots, x_m := N_m, x := x']])$

(by induction hypothesis)

$\equiv [\lambda\gamma.(\lambda x.M_1)[x_1 := N_1, \ldots, x_m := N_m]]$.

Case 4. $M \equiv cM_1 \ldots M_s$ (c is an s-ary constant).

$[\lambda\alpha.cM_1 \ldots M_s] \circ \langle[\lambda\gamma.N_1], \ldots, [\lambda\gamma.N_m]\rangle$

$\equiv (c \circ \langle[\lambda\alpha.M_1], \ldots, [\lambda\alpha.M_s]\rangle) \circ \langle[\lambda\gamma.N_1], \ldots, [\lambda\gamma.N_m]\rangle$

$\rightarrow c \circ \langle[\lambda\alpha.M_1] \circ \langle[\lambda\gamma.N_1], \ldots, [\lambda\gamma.N_m]\rangle, \ldots, [\lambda\alpha.M_s] \circ \langle[\lambda\gamma.N_1], \ldots, [\lambda\gamma.N_m]\rangle\rangle$

$\xrightarrow{*} c \circ \langle[\lambda\gamma.M_1[x_1 := N_1, \ldots, x_m := N_m]], \ldots, [\lambda\gamma.M_s[x_1 := N_1, \ldots, x_m := N_m]]\rangle$

(by induction hypothesis)

$\equiv [\lambda\gamma.cM_1 \ldots M_s[x_1 := N_1, \ldots, x_m := N_m]]$.

Case 5. $M \equiv M_1M_2$, and M is not of the form of case 4.

$[\lambda\alpha.M_1M_2] \circ \langle[\lambda\gamma.N_1], \ldots, [\lambda\gamma.N_m]\rangle$

$\equiv (\mathit{App} \circ \langle[\lambda\alpha.M_1], [\lambda\alpha.M_2]\rangle) \circ \langle[\lambda\gamma.N_1], \ldots, [\lambda\gamma.N_m]\rangle$

$\rightarrow \mathit{App} \circ \langle[\lambda\alpha.M_1] \circ \langle[\lambda\gamma.N_1], \ldots, [\lambda\gamma.N_m]\rangle, [\lambda\alpha.M_2] \circ \langle[\lambda\gamma.N_1], \ldots, [\lambda\gamma.N_m]\rangle\rangle$

$\xrightarrow{*} \mathit{App} \circ \langle[\lambda\gamma.M_1[x_1 := N_1, \ldots, x_m := N_m]], [\lambda\gamma.M_2[x_1 := N_1, \ldots, x_m := N_m]]\rangle$.

(by induction hypothesis)

Now, when $M_1M_2[x_1 := N_1, \ldots, x_m := N_m]$ is not of the form $cL_1 \ldots L_s$ (c is an s-ary constant), the last term is identical with $[\lambda\gamma.M_1M_2[x_1 := N_1, \ldots, x_m := N_m]]$. Suppose that $M_1M_2[x_1 := N_1, \ldots, x_m := N_m]$ is $cL_1 \ldots L_s$. Then $M_2[x_1 := N_1, \ldots, x_m := N_m]$ is L_s. The above last term is:

$\equiv \mathit{App} \circ \langle[\lambda\gamma.cL_1 \ldots L_{s-1}], [\lambda\gamma.L_s]\rangle$

$\xrightarrow{*} \mathit{App} \circ \langle\Lambda(c \circ \langle[\lambda\gamma.L_1] \circ \pi_1^{n,1}, \ldots, [\lambda\gamma.L_{s-1}] \circ \pi_1^{n,1}, p_{n+1}^{n+1}\rangle), [\lambda\gamma.L_s]\rangle$

(by Proposition 5.1)

$\rightarrow (c \circ \langle[\lambda\gamma.L_1] \circ \pi_1^{n,1}, \ldots, [\lambda\gamma.L_{s-1}] \circ \pi_1^{n,1}, p_{n+1}^{n+1}\rangle) \circ \langle p_1^n, \ldots, p_n^n, [\lambda\gamma.L_s]\rangle$

$\xrightarrow{*} c \circ \langle[\lambda\gamma.L_1], \ldots, [\lambda\gamma.L_{s-1}], [\lambda\gamma.L_s]\rangle$

$$\equiv [\lambda\gamma.cL_1 \dots L_{s-1}L_s].$$

6. Relationship between $CCLM_\beta$ and lambda calculus.

Now we are in a position to state the theorems which describe the relationship between the terms and reductions of the two systems $CCLM_\beta$ and λ, in terms of the two translation algorithms of the previous section.

Theorem 6.1. *Let $\gamma = \langle z_1, \dots, z_n \rangle$. Let M and N be terms in λ, and the free variables in them are all in γ. If $M \xrightarrow{*}_\lambda N$, then $[\lambda\gamma.M] \xrightarrow{*}_c [\lambda\gamma.N]$.*

Proof. By induction on the definition of $M \xrightarrow{*}_\lambda N$.

Case 1. α-rule.

$$\begin{aligned}
&[\lambda\gamma.(\lambda x.M_1)] \\
&\equiv \Lambda_{n+1}([\lambda\gamma x'.M_1[x := x']]) \\
&\equiv \Lambda_{n+1}([\lambda\gamma y'.M_1[x := y']]) && \text{(by Lemma 5.2)} \\
&\equiv [\lambda\gamma.(\lambda y.M_1[x := y])].
\end{aligned}$$

Case 2. β-rule.

$$\begin{aligned}
&[\lambda\gamma.(\lambda x.M_1)M_2] \\
&\equiv App \circ \langle \Lambda_{n+1}([\lambda\gamma x'.M_1[x := x']]), [\lambda\gamma.M_2] \rangle \\
&\rightarrow [\lambda\gamma x'.M_1[x := x']] \circ \langle p_1^n, \dots, p_n^n, [\lambda\gamma.M_2] \rangle \text{ (by rule 5)} \\
&\equiv [\lambda\gamma x'.M_1[x := x']] \circ \langle [\lambda\gamma.z_1], \dots, [\lambda\gamma.z_n], [\lambda\gamma.M_2] \rangle \\
&\xrightarrow{*} [\lambda\gamma.M_1[x := M_2]]. && \text{(by Lemma 5.4)}
\end{aligned}$$

Case 3. $M \equiv \lambda x.M_1$, $N \equiv \lambda x.N_1$, and $M_1 \xrightarrow{*} N_1$.

$$\begin{aligned}
&[\lambda\gamma.(\lambda x.M_1)] \\
&\equiv \Lambda([\lambda\gamma x'.M_1[x := x']]) \\
&\xrightarrow{*} \Lambda([\lambda\gamma x'.N_1[x := x']]) && \text{(by induction hypothesis)} \\
&\equiv [\lambda\gamma.(\lambda x.N_1)].
\end{aligned}$$

Case 4. $M \equiv M_1M_2$, $N \equiv N_1N_2$, $M_1 \xrightarrow{*} N_1$, and $M_2 \xrightarrow{*} N_2$.

(4-1) Suppose that M is of the form $cL_1 \dots L_s$, where c is an s-ary constant. Then N is also of the form $cL'_1 \dots L'_s$, and $L_1 \xrightarrow{*} L'_1, \dots, L_s \xrightarrow{*} L'_s$. Therefore,

$[\lambda\gamma.M_1M_2]$
$\equiv c \circ \langle [\lambda\gamma.L_1], \ldots, [\lambda\gamma.L_s] \rangle$
$\xrightarrow{*} c \circ \langle [\lambda\gamma.L'_1], \ldots, [\lambda\gamma.L'_s] \rangle$ (by induction hypothesis)
$\equiv [\lambda\gamma.N_1N_2]$.

(4-2) Suppose that M is not of the form in (4-1) and that N is of the form $cL_1 \ldots L_s$. Then $N_1 \equiv cL_1 \ldots L_{s-1}$ and $N_2 \equiv L_s$.

$[\lambda\gamma.M_1M_2]$
$\equiv App \circ \langle [\lambda\gamma.M_1], [\lambda\gamma.M_2] \rangle$
$\xrightarrow{*} App \circ \langle [\lambda\gamma.cL_1 \ldots L_{s-1}], [\lambda\gamma.L_s] \rangle$ (by induction hypothesis)
$\xrightarrow{*} [\lambda\gamma.cL_1 \ldots L_{s-1}L_s] \rangle$.

The last reduction has been shown in the calculation of case 5 of Lemma 5.4.

(4-3) Otherwise.

$[\lambda\gamma.M_1M_2]$
$\equiv App \circ \langle [\lambda\gamma.M_1], [\lambda\gamma.M_2] \rangle$
$\xrightarrow{*} App \circ \langle [\lambda\gamma.N_1], [\lambda\gamma.N_2] \rangle$ (by induction hypothesis)
$\equiv [\lambda\gamma.N_1N_2]$.

Theorem 6.2. *Let F and G be in $CCLM_\beta$, and both are n-ary. If $F \xrightarrow{*}_c G$ then*

$$F^*[N_1, \ldots, N_n] \xrightarrow{*}_\lambda G^*[N_1, \ldots, N_n].$$

Proof. By induction on the definition of $F \xrightarrow{*}_c G$.

Case 1. $F \equiv (H \circ \langle I_1, \ldots, I_l \rangle) \circ \langle J_1, \ldots, J_m \rangle$
$\to G \equiv H \circ \langle I_1 \circ \langle J_1, \ldots, J_m \rangle, \ldots, I_l \circ \langle J_1, \ldots, J_m \rangle \rangle$.

$F^*[N_1, \ldots, N_n]$
$\equiv H^*[I_1^*[J_1^*[N_1, \ldots, N_n], \ldots, J_m^*[N_1, \ldots, N_n]], \ldots,$
$I_l^*[J_1^*[N_1, \ldots, N_n], \ldots, J_m^*[N_1, \ldots, N_n]]]$
$\equiv G^*[N_1, \ldots, N_n]$.

Case 2. $F \equiv p_i^m \circ \langle H_1, \ldots, H_m \rangle \to G \equiv H_i$.

$F^*[N_1, \ldots, N_n]$
$\equiv (p_i^m)^*[H_1^*[N_1, \ldots, N_n], \ldots, H_m^*[N_1, \ldots, N_n]]$
$\equiv G^*[N_1, \ldots, N_n]$.

Case 3. $F \equiv H \circ \langle p_1^n, \ldots, p_n^n \rangle \to G \equiv H$.

$F^*[N_1, \ldots, N_n]$
$\equiv H^*[(p_1^n)^*[N_1, \ldots, N_n], \ldots, (p_n^n)^*[N_1, \ldots, N_n]]$
$\equiv G^*[N_1, \ldots, N_n]$.

Case 4. $F \equiv \Lambda_{m+1}(H) \circ \langle I_1, \ldots, I_m \rangle \to G \equiv \Lambda_{n+1}(H \circ \langle I_1 \circ \pi_1^{n,1}, \ldots, I_m \circ \pi_1^{n,1}, p_{n+1}^{n+1} \rangle)$.

$F^*[N_1, \ldots, N_n]$

$\equiv (\Lambda(H))^*[I_1^*[N_1, \ldots, N_n], \ldots, I_m^*[N_1, \ldots, N_n]]$
$\equiv \lambda x.H^*[I_1^*[N_1, \ldots, N_n], \ldots, I_m^*[N_1, \ldots, N_n], x]$
$\equiv G^*[N_1, \ldots, N_n]$.

Case 5. $F \equiv App \circ \langle \Lambda_{n+1}(H), I \rangle \to G \equiv H \circ \langle p_1^n, \ldots, p_n^n, I \rangle$.
$F^*[N_1, \ldots, N_n]$
$\equiv (\lambda x.H^*[N_1, \ldots, N_n, x])(I^*[N_1, \ldots, N_n])$
$\to_\lambda H^*[N_1, \ldots, N_n, I^*[N_1, \ldots, N_n]]$ (by β-rule)
$\equiv G^*[N_1, \ldots, N_n]$.

Case 6. $F \equiv H \circ \langle I_1, \ldots, I_m \rangle$, $G \equiv H' \circ \langle I_1', \ldots, I_m' \rangle$, $H \xrightarrow{*} H'$, $I_1 \xrightarrow{*} I_1', \ldots, I_m \xrightarrow{*} I_m'$.
$F^*[N_1, \ldots, N_n]$
$\equiv H^*[I_1^*[N_1, \ldots, N_n], \ldots, I_m^*[N_1, \ldots, N_n]]$
$\xrightarrow{*} (H')^*[(I_1')^*[N_1, \ldots, N_n], \ldots, (I_m')^*[N_1, \ldots, N_n]]$ (by induction hypothesis)
$\equiv G^*[N_1, \ldots, N_n]$.

Case 7. $F \equiv \Lambda(H)$, $G \equiv \Lambda(H')$, and $H \xrightarrow{*} H'$.
By induction hypothesis.

Before going into the next Theorem 6.3, we need the following definition.

Definition. For a term F in $CCLM_\beta$, let F^+ be the term in $CCLM_\beta$ obtained by replacing all n-ary constants f (including p_i^n and App) in F by $f \circ \langle p_1^n, \ldots, p_n^n \rangle$, for all $n \geq 0$.

Theorem 6.3. *Let F be of arity n, and $\gamma \equiv \langle z_1, \ldots, z_n \rangle$. Then we have*

$$F^+ \xrightarrow{*}_c [\lambda\gamma.F^*[z_1, \ldots, z_n]].$$

Proof. By induction on the structure of F.

Case 1. $F \equiv p_i^n$.
$(p_i^n)^+$
$\equiv p_i^n \circ \langle p_1^n, \ldots, p_n^n \rangle$
$\to p_i^n$
$\equiv [\lambda\gamma.(p_i^n)^*[z_1, \ldots, z_n]]$.

Case 2. $F \equiv App$.
App^+
$\equiv App \circ \langle p_1^2, p_2^2 \rangle$
$\equiv [\lambda\langle z_1, z_2 \rangle.z_1 z_2]$
$\equiv [\lambda\langle z_1, z_2 \rangle.App^*[z_1, z_2]]$.

Case 3. $F \equiv f$, an n-ary constant other than p_i^n and App.
f^+

$\equiv f \circ \langle p_1^n, \ldots, p_n^n \rangle$
$\equiv f \circ \langle [\lambda\gamma.z_1], \ldots, [\lambda\gamma.z_n] \rangle$
$\equiv [\lambda\gamma.f z_1 \ldots z_n]$
$\equiv [\lambda\gamma.f^*[z_1, \ldots, z_n]]$.

Case 4. $F \equiv H \circ \langle I_1, \ldots, I_m \rangle$.
$(H \circ \langle I_1, \ldots, I_m \rangle)^+$
$\equiv H^+ \circ \langle I_1^+, \ldots, I_m^+ \rangle$
$\xrightarrow{*} [\lambda\alpha.H^*[x_1, \ldots, x_m]] \circ \langle [\lambda\gamma.I_1^*[z_1, \ldots, z_n]], \ldots, [\lambda\gamma.I_m^*[z_1, \ldots, z_n]] \rangle$
(by induction hypothesis)
$\xrightarrow{*} [\lambda\gamma.H^*[I_1^*[z_1, \ldots, z_n], \ldots, I_m^*[z_1, \ldots, z_n]]]$ (by Lemma 5.4)
$\equiv [\lambda\gamma.(H \circ \langle I_1, \ldots, I_m \rangle)^*[z_1, \ldots, z_n]]$.

Case 5. $F \equiv \Lambda(H)$.
$(\Lambda(H))^+$
$\equiv \Lambda(H^+)$
$\xrightarrow{*} \Lambda([\lambda\gamma z.H^*[z_1, \ldots, z_n, z]])$ (by induction hypothesis)
$\equiv [\lambda\gamma.(\lambda z.H^*[z_1, \ldots, z_n, z])]$
$\equiv [\lambda\gamma.(\Lambda(H))^*[z_1, \ldots, z_n]]$.

For Theorem 6.4 we need the following definition.

Definition. For a term M in λ let M° be the term obtained from M by replacing all occurrences of s-ary constants c not appearing in the form $cM_1 \ldots M_s$ by $\lambda x_1 \ldots \lambda x_s.cx_1 \ldots x_s$.

Let $M \equiv_\lambda N$ mean that M and N are syntactically identical except bound variables.

Theorem 6.4. *Let M in λ, $\gamma \equiv \langle z_1, \ldots, z_n \rangle$, and all the free variables in M are in γ. Then we have*

$$[\lambda\gamma.M]^*[z_1, \ldots, z_n] \equiv_\lambda M^\circ.$$

Proof. Induction by the structure of M.

Case 1. $M \equiv z_i \quad (1 \le i \le n)$.
$[\lambda\gamma.z_i]^*[z_1, \ldots, z_n]$
$\equiv (p_i^n)^*[z_1, \ldots, z_n]$
$\equiv z_i$.

Case 2. $M \equiv c \quad$ (s-ary constant).
When $s = 0$ it is easy. Assume $s \ge 1$.

$$
\begin{aligned}
&[\lambda\gamma.c]^*[z_1,\ldots,z_n] \\
&\equiv (\Lambda^s(c \circ \pi_2^{n,s}))^*[z_1,\ldots,z_n] \\
&\equiv \lambda x_1 \ldots \lambda x_s.((c \circ \pi_2^{n,s})^*[z_1,\ldots,z_n,x_1,\ldots,x_s]) \\
&\equiv \lambda x_1 \ldots \lambda x_s.cx_1 \ldots x_s.
\end{aligned}
$$

Case 3. $M \equiv \lambda x.M_1$.

$$
\begin{aligned}
&[\lambda\gamma.(\lambda x.M_1)]^*[z_1,\ldots,z_n] \\
&\equiv (\Lambda_{n+1}([\lambda\gamma x'.M_1[x := x']]))^*[z_1,\ldots,z_n] \\
&\equiv \lambda y.([\lambda\gamma x'.M_1[x := x']]^*[z_1,\ldots,z_n,y]) \\
&\equiv_\lambda \lambda y.M_1^\circ[x := y]. && \text{(by induction hypothesis)}
\end{aligned}
$$

Case 4. $M \equiv cM_1 \ldots M_s$ (c is an s-ary constant).

$$
\begin{aligned}
&[\lambda\gamma.cM_1 \ldots M_s]^*[z_1,\ldots,z_n] \\
&\equiv (c \circ \langle[\lambda\gamma.M_1],\ldots,[\lambda\gamma.M_s]\rangle)^*[z_1,\ldots,z_n] \\
&\equiv c([\lambda\gamma.M_1]^*[z_1,\ldots,z_n])\ldots([\lambda\gamma.M_s]^*[z_1,\ldots,z_n]) \\
&\equiv_\lambda cM_1^\circ \ldots M_s^\circ. && \text{(by induction hypothesis)}
\end{aligned}
$$

Case 5. $M \equiv M_1M_2$, and not of the form of case 4.

$$
\begin{aligned}
&[\lambda\gamma.M_1M_2]^*[z_1,\ldots,z_n] \\
&\equiv (App \circ \langle[\lambda\gamma.M_1],[\lambda\gamma.M_2]\rangle)^*[z_1,\ldots,z_n] \\
&\equiv ([\lambda\gamma.M_1]^*[z_1,\ldots,z_n])([\lambda\gamma.M_2]^*[z_1,\ldots,z_n]) \\
&\equiv_\lambda M_1^\circ M_2^\circ. && \text{(by induction hypothesis)}
\end{aligned}
$$

7. Church-Rosser property.

In this section we prove the confluency property for $CCLM_\beta$.

Definition. For a term F in $CCLM_\beta$ define the term F^- in $CCLM_\beta$ as follows.
1. For a constant f, including p_i^n and App, $f^- \equiv f$.

2. $(G \circ \langle H_1, \ldots, H_n \rangle)^- \equiv \begin{cases} H_i^- & \text{if } G^- \equiv p_i^n, \\ G^- & \text{if } \langle H_1^-, \ldots, H_n^- \rangle \equiv \langle p_1^n, \ldots, p_n^n \rangle, \\ G^- \circ \langle H_1^-, \ldots, H_n^- \rangle & \text{otherwise.} \end{cases}$

3. $(\Lambda(G))^- \equiv \Lambda(G^-)$.

Thus, F^- is the term obtained from F by applying reductions concerning projections (rules 2 and 3) as much as possible. We state without proof easy properties of this transformation.

Lemma 7.1.

i) $F \xrightarrow{*}_c F^-$.
ii) $(F^-)^- \equiv F^-$.
iii) $F \xrightarrow{*}_c (F^+)^-$.

Lemma 7.2. $F \xrightarrow{*}_c G$ *then* $F^- \xrightarrow{*}_c G^-$.

Proof. By induction on the length of reduction of $F \xrightarrow{*} G$.
Case 1. $F \equiv (H \circ \langle I_1, \ldots, I_m \rangle) \circ \langle J_1, \ldots, J_n \rangle \to G \equiv H \circ \langle I_1 \circ \langle J_1, \ldots, J_n \rangle, \ldots \rangle$.
(1-1) $H^- \equiv p_i^m$.
$$F^- \equiv (I_i \circ \langle J_1, \ldots, J_n \rangle)^- \equiv G^-.$$
(1-2) $\langle I_1^-, \ldots, I_m^- \rangle \equiv \langle p_1^m, \ldots, p_m^m \rangle$ (necessarily $m = n$).
$$F^- \equiv (H \circ \langle J_1, \ldots, J_n \rangle)^- \equiv G^-.$$
(1-3) $\langle J_1^-, \ldots, J_n^- \rangle \equiv \langle p_1^n, \ldots, p_n^n \rangle$.
$$F^- \equiv (H \circ \langle I_1, \ldots, I_m \rangle)^- \equiv G^-.$$
(1-4) Otherwise.
$$F^- \equiv (H^- \circ \langle I_1^-, \ldots, I_m^- \rangle) \circ \langle J_1^-, \ldots, J_n^- \rangle \to H^- \circ \langle I_1^- \circ \langle J_1^-, \ldots, J_n^- \rangle, \ldots \rangle \equiv G^-.$$
Case 2. $F \equiv p_i^n \circ \langle H_1, \ldots, H_n \rangle \to G \equiv H_i$.
$$F^- \equiv H_i^- \equiv G^-.$$
Case 3. $F \equiv H \circ \langle p_1^n, \ldots, p_n^n \rangle \to G \equiv H$.
$$F^- \equiv H^- \equiv G^-.$$
Case 4. $F \equiv \Lambda(H) \circ \langle I_1, \ldots, I_n \rangle^k \to G \equiv \Lambda(H \circ \langle I_1 \circ \pi_1^{k,1}, \ldots, I_n \circ \pi_1^{k,1}, p_{k+1}^{k+1} \rangle)$.
(4-1) $\langle I_1^-, \ldots, I_n^- \rangle \equiv \langle p_1^n, \ldots, p_n^n \rangle$.
$$F^- \equiv \Lambda(H^-) \equiv G^-.$$

(4-2) $H^- \equiv p_i^{n+1}$, and not of the form of (4-1).

$$\begin{aligned}F^- &\equiv \Lambda(H^-) \circ \langle I_1^-, \ldots, I_n^- \rangle \\ &\to \Lambda(p_i^{n+1} \circ \langle I_1^- \circ \pi_1^{k,1}, \ldots, I_n^- \circ \pi_1^{k,1}, p_{k+1}^{k+1} \rangle) \\ &\to G^-.\end{aligned}$$

(4-3) Otherwise.

$$\begin{aligned}F^- &\equiv \Lambda(H^-) \circ \langle I_1^-, \ldots, I_n^- \rangle \\ &\to \Lambda(H^- \circ \langle I_1^- \circ \pi_1^{k,1}, \ldots, I_n^- \circ \pi_1^{k,1}, p_{k+1}^{k+1} \rangle) \\ &\equiv G^-.\end{aligned}$$

Case 5. $F \equiv App \circ \langle \Lambda(H), I \rangle \to G \equiv H \circ \langle p_1^n, \ldots, p_n^n, I \rangle$.

(5-1) $H^- \equiv p_i^{n+1}$.

$$\begin{aligned}F^- &\equiv App \circ \langle \Lambda(H^-), I^- \rangle \\ &\to p_i^{n+1} \circ \langle p_1^n, \ldots, p_n^n, I^- \rangle \\ &\to G^-.\end{aligned}$$

(5-2) Otherwise.

$$\begin{aligned}F^- &\equiv App \circ \langle \Lambda(H^-), I^- \rangle \\ &\to H^- \circ \langle p_1^n, \ldots, p_n^n, I^- \rangle \\ &\equiv G^-.\end{aligned}$$

Case 6. $F \equiv H \circ \langle I_1, \ldots, I_n \rangle \xrightarrow{*} G \equiv H' \circ \langle I_1', \ldots, I_n' \rangle$, $H \xrightarrow{*} H'$, and $I_1 \xrightarrow{*} I_1', \ldots, I_n \xrightarrow{*} I_n'$.

(6-1) $H \equiv p_i^n$.

Since $H' \equiv p_i^n$, by induction hypothesis we have

$$F^- \equiv I_i^- \xrightarrow{*} (I_i')^- \equiv G^-.$$

(6-2) $\langle I_1, \ldots, I_n \rangle \equiv \langle p_1^n, \ldots, p_n^n \rangle$.

Since $\langle I_1', \ldots, I_n' \rangle \equiv \langle p_1^n, \ldots, p_n^n \rangle$, by induction hypothesis we have

$$F^- \equiv H^- \xrightarrow{*} (H')^- \equiv G^-.$$

(6-3) Otherwise.

$$\begin{aligned}F^- &\equiv H^- \circ \langle I_1^-, \ldots, I_n^- \rangle \\ &\xrightarrow{*} (H')^- \circ \langle (I_1')^-, \ldots, (I_n')^- \rangle && \text{(by induction hypothesis)} \\ &\xrightarrow{*} G^-.\end{aligned}$$

Case 7. $F \equiv \Lambda(H) \xrightarrow{*} G \equiv \Lambda(H')$ and $H \xrightarrow{*} H'$.

By induction hypothesis.

The following theorem establishes the Church-Rosser property of $CCLM_\beta$.

Theorem 7.3. *If $F \xrightarrow{*}_c G_1$, and $F \xrightarrow{*}_c G_2$, then there exists H such that $G_1 \xrightarrow{*}_c H$ and $G_2 \xrightarrow{*}_c H$.*

Proof. Suppose

$$F \xrightarrow{*}_c G_1, \text{ and } F \xrightarrow{*}_c G_2.$$

Assume that F is n-ary, so that G_1 and G_2 are also n-ary. By Theorem 6.2 we have

$$F^*[z_1, \ldots, z_n] \xrightarrow{*}_\lambda G_1^*[z_1, \ldots, z_n], \text{ and } F^*[z_1, \ldots, z_n] \xrightarrow{*}_\lambda G_2^*[z_1, \ldots, z_n].$$

By the Church-Rosser theorem of the type-free λ_β-calculus, there exists M such that

$$G_1^*[z_1, \ldots, z_n] \xrightarrow{*}_\lambda M, \text{ and } G_2^*[z_1, \ldots, z_n] \xrightarrow{*}_\lambda M.$$

Let $\gamma \equiv \langle z_1, \ldots, z_n \rangle$. By Theorem 6.1 we have

$$[\lambda\gamma.G_1^*[z_1, \ldots, z_n]] \xrightarrow{*}_c [\lambda\gamma.M], \text{ and } [\lambda\gamma.G_2^*[z_1, \ldots, z_n]] \xrightarrow{*}_c [\lambda\gamma.M]. \quad (1)$$

By the way, by Theorem 6.3 we have

$$G_1^+ \xrightarrow{*}_c [\lambda\gamma.G_1^*[z_1, \ldots, z_n]], \text{ and } G_2^+ \xrightarrow{*}_c [\lambda\gamma.G_2^*[z_1, \ldots, z_n]]. \quad (2)$$

By combining (1) and (2) we get

$$G_1^+ \xrightarrow{*}_c [\lambda\gamma.M], \text{ and } G_2^+ \xrightarrow{*}_c [\lambda\gamma.M].$$

Applying Lemma 7.2 to the above we have

$$(G_1^+)^- \xrightarrow{*}_c [\lambda\gamma.M]^-, \text{ and } (G_2^+)^- \xrightarrow{*}_c [\lambda\gamma.M]^-.$$

Therefore, if we put H as $[\lambda\gamma.M]^-$, by using Lemma 7.1 iii) we have

$$G_1 \xrightarrow{*}_c (G_1^+)^- \xrightarrow{*}_c H, \text{ and } G_2 \xrightarrow{*}_c (G_2^+)^- \xrightarrow{*}_c H.$$

The proof is completed.

Remark. We have reduced the proof of Church-Rosser property of $CCLM_\beta$ to that of lambda calculus by using the relationship of the reductions between the two systems. A direct proof, that is, a proof solely within the system, may be possible, but it will perhaps be a tedious one. For the CCM calculus of Yokouchi, a direct proof of the property is in [14].

REFERENCES

[1] J. Backus, Can programming be liberated from the von Neumann style? A functional style and its algebra of programs, Comm. ACM, 21 (1978), pp. 613–641.

[2] H. P. Barendregt, The Lambda Calculus: Its Syntax and Semantics, Revised ed., North-Holland, Amsterdam, 1984.

[3] G. Cousineau, P.-L. Curien and M. Mauny, The categorical abstract machine, Sci. Comput. Program., 8 (1987), pp. 173–202.

[4] P.-L. Curien, Categorical combinators, Inform. Contr., 69 (1986), pp. 188–254.

[5] P.-L. Curien, Categorical Combinators, Sequential Algorithms and Functional Programming, Pitman, London, 1986.

[6] Y. Futamura, Partial computation of programs, Lecture Notes in Computer Science, vol. 147, pp. 1–35, Springer-Verlag, Berlin, 1983.

[7] S. Hayashi, Adjunction of semifunctors: Categorical structures in nonextensional lambda calculus, Theoret. Comput. Sci., 41 (1985), pp. 95–104.

[8] C. P. J. Koymans, Models of the lambda calculus, Inform. Contr., 52 (1982), pp. 306–332.

[9] J. Lambek, Functional completeness of cartesian categories, Ann. Math. Logic, 6 (1974), pp. 259–292.

[10] J. Lambek and P. J. Scott, Introduction to Higher Order Categorical Logic, Cambridge University Press, Cambridge, 1986.

[11] A. R. Meyer, What is a model of the lambda calculus?, Inform. Contr., 52 (1982), pp. 87–122.

[12] H. Yokouchi, Application and composition in functional programming, J. Inform. Process., 8 (1985), pp. 217–221.

[13] H. Yokouchi, Retraction map categories and their applications to the construction of lambda calculus models, Inform. Contr., 71 (1986), pp. 33–86.

[14] H. Yokouchi, The CCM calculus — another formalization of the λ-calculus, preprint, 1987.